

Everything is a Prompt

A General Recursive Lag–Gradient Framework for Complex Systems

Luke Doel

March 22, 2026

License: Creative Commons Zero (CC0) – Public Domain Dedication

Abstract

We propose a general dynamical framework in which systems evolve through recursive state updates driven by delayed responses to prior states. In this formulation, all observable structures arise from a single mechanism:

systems respond to inputs with lag; lag produces gradients; gradients produce behaviour; and behaviour becomes the next input state.

We refer to this as a recursive lag–gradient loop (RLG loop). We show that this structure can be consistently mapped onto physical, computational, and biological domains, providing a unified description of emergent complexity without requiring domain-specific mechanisms at the level of the base model.

We identify conditions under which the framework produces scale-dependent structure formation and discuss its relationship to known delayed dynamical systems.

1 Introduction

Many complex systems exhibit emergent behaviour that is not trivially reducible to instantaneous local interactions. These include:

- gravitational structure formation
- neural computation
- evolutionary adaptation
- self-organising computational systems

Despite differing substrates, these systems share a common feature: non-instantaneous response to prior states.

We propose a unifying description: systems do not update instantaneously; they evolve through lagged responses to prior configurations. This lag induces structural asymmetries (gradients), which generate observable dynamics.

2 Core Hypothesis

We define the central mechanism:

Postulate 1 (Recursive Prompting of Systems)

All systems evolve according to:

$\text{State}_t \rightarrow \text{Lagged Response} \rightarrow \text{Gradient Formation} \rightarrow \text{Behaviour} \rightarrow \text{State}_{t+1}$

or compactly:

$$S_{t+1} = \mathcal{F}(S_t, \tau(S_t)) \quad (1)$$

where:

- S_t is system state
- τ is a state-dependent lag function
- $\mathcal{F}$ is a response operator

3 Interpretation of Components

3.1 State (Prompt)

The system configuration at a given time. This may represent:

- mass-energy distribution (physics)
- neuron activation patterns (cognition)
- population structure (evolution)
- computational grid state (CA systems)

3.2 Lag

Lag is defined as the finite delay in system response to state change:

$$\tau = f(S) \quad (2)$$

Lag is not treated as noise but as a generative parameter of structure.

3.3 Gradient Formation

Non-uniform lag induces spatial or informational asymmetries:

$$\nabla \tau(S) \quad (3)$$

These gradients define directional tendencies in system evolution.

3.4 Behaviour

Behaviour is the system's response to gradients:

$$B \propto \nabla \tau(S) \quad (4)$$

This corresponds to motion, adaptation, or rule application depending on substrate.

3.5 Recursive Closure

Behaviour updates system state:

$$S_{t+1} = S_t + B \quad (5)$$

This closes the loop.

4 Cross-Domain Mapping

4.1 Physical systems

- State: field configuration
- Lag: finite propagation / interaction delay
- Gradient: spatial variation in response fields
- Behaviour: motion / curvature of trajectories

Interpretation: structure arises from asynchronous response across fields

4.2 Computational systems

- State: grid configuration
- Lag: discrete update step
- Gradient: local state differences
- Behaviour: rule execution

Interpretation: computation is explicit instantiation of the loop

4.3 Biological evolution

- State: population distribution
- Lag: generational delay
- Gradient: fitness differentials
- Behaviour: selection and adaptation

Interpretation: evolution is a slow recursive lag process across generations

5 Invariant Claim

Across all domains: system behaviour is determined not by static rules alone, but by how state-dependent lag structures generate gradients over time.

6 Emergent Properties

The framework implies:

6.1 Structure as frozen recursion

Observed structures represent accumulated states of repeated loop iterations.

6.2 Time as iteration depth

Time corresponds to the number of recursive lag updates applied to a system.

6.3 Complexity from delay

Increased lag heterogeneity increases structural complexity.

7 Falsifiable Predictions

The framework implies:

1. Reducing effective lag should reduce emergent structural complexity.
2. Systems with uniform lag should not generate stable structure.
3. Increasing lag heterogeneity should increase pattern formation.
4. Cross-domain systems with analogous lag structure should exhibit analogous emergent behaviour.

8 Limitations

This formulation currently lacks:

- a fully specified functional form for $\tau(S)$
- stability constraints for recursive iteration
- separation of gradient types across domains
- derivation of specific physical laws from first principles

It should therefore be considered: a general structural hypothesis rather than a complete physical theory.

9 Discussion

This framework does not claim to replace existing theories in physics, computation, or biology.

Instead, it proposes: a unifying description of how delayed response systems generate structure through recursive feedback.

It may be interpreted as a higher-level constraint on how complex systems evolve rather than a replacement for domain-specific laws.

10 Conclusion

We propose that: recursive lag in state-dependent systems is sufficient to generate gradients, and gradients are sufficient to generate behaviour, which recursively defines subsequent system states.

This simple loop may underlie a broad class of emergent phenomena across multiple domains.